



1) introduction

Nous allons nous intéresser à un acteur fondamental du développement web, le couple HTML+CSS (Hyper Text Markup Language et Cascading Style Sheets).

2) le HTML

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

Pour l'instant, nous allons retenir deux éléments de cette définition «conçu pour représenter les pages web» et «un langage de balisage».

Grâce au HTML vous allez pouvoir, dans votre navigateur (Firefox, Chrome, Opera,...), afficher du texte, afficher des images, proposer des hyperliens (liens vers d'autres pages web), afficher des formulaires et même maintenant afficher des vidéos (grâce à la dernière version du HTML, l'HTML5).

HTML n'est pas un langage de programmation (comme le Python par exemple), ici, pas question de conditions, de boucles...c'est un langage de description.

Voici un exemple de code HTML :

```
<h1>Hello World! Ceci est un titre</h1>
<p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

Comme déjà évoqué ci-dessus, en HTML tout est une histoire de balise que l'on ouvre et que l'on ferme. Une balise ouvrante est de la forme `<nom_de_la_balise>` , les balises fermantes sont de la forme `</nom_de_la_balise>` .

En observant attentivement le code, vous devriez forcément remarquer que toute balise ouverte doit être refermée à un moment ou un autre. La balise ouvrante et la balise fermante peuvent être sur la même ligne ou pas, cela n'a aucune espèce d'importance, la seule question à se poser ici est : ai-je bien refermé toutes les balises que j'ai ouvertes ?

Enfin pour terminer avec les généralités sur les balises, il est important de savoir qu'une structure du type :

```
<balise1>
<balise2>
</balise1>
</balise2>
```

est interdite, la balise2 a été ouverte après la balise1, elle devra donc être refermée avant la balise1.

En revanche, l'enchaînement suivant est correct :

```
<balise1>
<balise2>
</balise2>
</balise1>
```

Notez que dans notre exemple nous respectons bien cette règle « d'imbrication » des balises avec la balise `<p>` et la balise `` .

Il est important de comprendre que chaque balise a une signification qu'il faut bien respecter (on parle de la sémantique des balises). Par exemple le texte situé entre la balise ouvrante et fermante `<h1>` est obligatoirement un titre important (il existe des balises `<h2>` , `<h3>` ...qui sont aussi des titres, mais des titres moins importants (sous-titre)). La balise `<p>` permet de définir des paragraphes, enfin, la balise `` permet de mettre en évidence un élément important.

Vous devez aussi savoir qu'il existe des balises qui sont à la fois ouvrantes et fermantes (`<balise/>`) : un exemple, la balise permettant de sauter une ligne, la balise `
` (balise qu'il faut d'ailleurs éviter d'utiliser par différentes raisons que nous n'aborderons pas ici).

Il est possible d'ajouter des éléments à une balise ouvrante, on parle d'attribut. Une balise peut contenir plusieurs attributs :

```
<ma_balise attribut_1= "valeur_1" attribut_2="valeur_2">
```

Il existe beaucoup d'attributs différents, nous allons nous contenter de 2 exemples avec l'attribut id (id pour identifiant) et class. Nous verrons l'intérêt de ces attributs prochainement

code HTML avec une balise `<p>` possédant un attribut id :

```
<h1>Ceci est un titre</h1>
<h2 class="titre_1">Ceci est un sous titre</h2>
<p id="para_1">Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

3) le CSS

Le HTML n'a pas été conçu pour gérer la mise en page (c'est possible, mais c'est une mauvaise pratique). Le HTML s'occupe uniquement du contenu et de la sémantique, pour tout ce qui concerne la mise en page et l'aspect "décoratif" (on parle du « style » de la page), on utilisera le CSS (Cascading Style Sheets).

Voici un exemple de code CSS :

```
h1
{
  text-align: center;
  background-color: red;
}
h2
{
  font-family: Verdana;
  font-style: italic;
  color: green;
}
```

Dans l'exemple ci-dessus, les propriétés *text-align* (permet de choisir l'alignement d'un texte) et *background-color* (permet de modifier la couleur de fond) seront appliquées au contenu de toutes les balises de type *h1* (avec respectivement les valeurs *center* et *red* : les textes contenus dans les balises *h1* seront centrés et la couleur de fond sera rouge).

Le contenu des balises *h2* sera écrit avec la police de caractère *Verdana*, en italique et en vert.

Si on désire cibler seulement certaines balises et pas les autres, il est possible d'utiliser les attributs *id* et *class*:

voici du code HTML :

```
<h1>Ceci est un titre</h1>
<p>Bonjour</p>
<p id="para_1">Vous allez bien ?</p>
```

et le code CSS correspondant :

```
#para_1
{
  font-style: italic;
  color: green;
}
```

Ici seul le second paragraphe (*Vous allez bien ?*) sera concerné par le CSS ci-dessus. Le premier paragraphe (*Bonjour*) et le titre *h1* ne seront pas concernés

Il est donc possible de cibler un paragraphe et pas un autre en utilisant l'id du paragraphe (en CSS l'id se traduisant par le signe #).

Il est aussi possible d'utiliser l'attribut class à la place de l'id. Dans le CSS on utilisera le point . à la place du #.

La différence entre "id" et "class" n'est pas très importante.

L'attribut "class" permet de donner le même nom à plusieurs reprises dans une même page, ce qui n'est pas possible avec l'attribut *id*

Si nous avions eu un 3e paragraphe, nous aurions pu avoir : `<p class="para_1">Voici un 3e paragraphe</p>` , mais nous n'aurions pas pu avoir : `<p id="para_1"> Voici un 3e paragraphe </p>` , car le nom para_1 a déjà été utilisé pour le 1er paragraphe.

4) quelques balises à connaître

a) la balise *a*

La balise *a* permet de créer des liens hypertextes, ce sont ces liens hypertextes qui vous permettent de "voyager" entre les pages d'un site ou entre les sites. Les liens hypertextes sont par défaut soulignés et de couleur bleue (modifiable grâce au CSS). La balise *a* possède un attribut *href* qui a pour valeur le chemin du fichier que l'on cherche à atteindre ou l'adresse du site cible :

```
<a href="https://pixees.fr">Cliquez ici pour vous rendre sur le site pixees</a>
```

Entre la balise ouvrante et fermante, on trouve le texte qui s'affichera à l'écran (c'est ce texte qui est souligné et de couleur bleue). La balise peut sans problème se trouver en plein milieu d'un paragraphe.

b) la balise *image*

Comme vous devez déjà vous en douter, la balise image sert à insérer des...images :

```

```

la balise `img` est à la fois ouvrante et fermante comme la balise `br`. Elle possède 2 attributs :

`src` qui doit être égal au nom du fichier image (ou au chemin si le fichier image se trouve dans un autre dossier).

`alt` qui doit être égal à une description de votre image (cet attribut est utilisé notamment par les systèmes de description des pages web utilisées par les non-voyants), il faut donc systématiquement renseigner cet attribut.

c) les balises `form`, `label`, `input` et `button`

Les formulaires sont des éléments importants des sites internet, ils permettent à l'utilisateur de transmettre des informations. Un formulaire devra être délimité par une balise `form` (même si ce n'est pas une obligation) :

```
<form>
....
</form>
```

Il existe différentes balises permettant de construire un formulaire, notamment la balise `input`. Cette balise possède un attribut `type` qui lui permet de jouer des rôles très différents.

Les balises `button` et `label` nous seront aussi d'une grande utilité :

```
<form>
  <label>voici un champ de texte : </label><input type="text"/>
  <button>Cliquez ici !</button>
</form>
```

d) les balises `div` et `span`

Ces 2 balises sont très utilisées (surtout la balise `div`). Pourtant, il faudrait, autant que possible, les éviter, pourquoi ?

Parce qu'elles n'ont aucune signification particulière, ce sont des balises dites "génériques".

À quoi servent-elles alors ?

À organiser la page, à regrouper plusieurs balises dans une même entité :

```
<div>
  <h2> Mon en-tête </h2>
  <p>Hello, comment ça va ?</p>
```

```
</div>
<div>
  <h2>mon <span>bas</span> de page</h2>
  <p>Au revoir !</p>
</div>
```

Pourquoi 2 balises génériques ?

Parce que div est une balise de type "block" et que span est une balise de type "inline" !

5) les balises de type *block* et les balises de type *inline*

Sans trop entrer dans les détails, il faut bien comprendre que l'ordre des balises dans le code HTML a une grande importance. Les balises sont affichées les unes après les autres, on parle du flux normal de la page.

C'est ici qu'entrent en jeu les balises de type "block" et les balises de type "inline".

Les contenus des balises de type "block" (p, div, h1, ...) se placent sur la page web les uns en dessous des autres.

Les contenus des balises de type "inline" (strong, img, span, a) se placent sur la page web les uns à côté des autres.

Cela revient à dire qu'une balise de type "block" prend toute la largeur de la page alors qu'une balise de type "inline" prend juste la largeur qui lui est nécessaire.

6) écrire une page HTML

Pour écrire une page HTML complète, il est nécessaire de rajouter quelques balises :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>

  </body>
</html>
```

Nous n'allons pas nous attarder sur ce squelette de page HTML, mais voici ce que vous devez savoir :

- la première ligne (`<!doctype html>`) permet d'indiquer au navigateur que nous utiliserons du HTML
- la balise *html* est obligatoire, l'attribut `lang="fr"` permet d'indiquer au navigateur que nous utiliserons le français pour écrire notre page.
- la balise *head* délimite ce que l'on appelle l'en-tête. L'en-tête contient, dans notre exemple, 3 balises :
 - la balise *meta* possède l'attribut `charset="utf-8"` qui permet de définir l'encodage des caractères (ici utf-8)
 - la balise *title* qui définit le titre de la page (attention ce titre ne s'affiche pas dans le navigateur, ne pas confondre avec la balise *h1*, on retrouve le contenu de cette balise au niveau des onglets du navigateur)
 - la balise *link* va permettre de relier votre HTML et votre CSS. Votre CSS se trouvera dans un fichier indépendant (ici *style.css*), l'attribut *href* correspond au chemin (relatif ou absolu) vers ce fichier
- toutes les balises qui composeront votre page (p, h1, img...) devront se trouver entre la balise *body* ouvrante et la balise *body* fermante.

Pour créer une nouvelle page, il suffira de copier-coller ce squelette de page HTML dans un nouveau fichier et de le compléter avec votre code HTML (entre les balises *body*)



activité 23.1

- créez un répertoire nommé *site_web* dans votre répertoire de travail
- créez un fichier *index.html* dans le répertoire *site_web*
- créez un fichier *style.css* dans le répertoire *site_web*
- créez un répertoire *image* dans le répertoire *site_web*
- copiez-collez l'image que vous téléchargerez [ici](#) dans le répertoire *image* que vous venez de créer
- copiez-collez le code HTML suivant dans le fichier *index.html*

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="style.css" />
  <title>Informatique lycée</title>
</head>

<body>
  <header>
    <nav>
      <a href="https://pixees.fr/informatiquelycee/sec/index.html" target="
      <a href="https://pixees.fr/informatiquelycee/prem/index.html" target=
      <a href="https://pixees.fr/informatiquelycee/term/index.html" target=
    </nav>
  </header>
  <div class="content">
    <h1>INFORMATIQUE LYCÉE</h1>
    <hr>
    <h2>SNT - NSI</h2>
  </div>
</body>

</html>
```

- copiez-collez le code CSS suivant dans le fichier *style.css*

```
@font-face {
  font-family: "cormorant";
  src: url("url('https://fonts.googleapis.com/css2?family=Cormorant+Garamond&di
```

```

}
* {
  margin: 0;
  padding: 0;
  font-family: Helvetica, Georgia, sans-serif;
}

/*****header*****/
header {
  background-color: rgb(42, 42, 42, 0.4);
  position: fixed;
  width: 100%;
  height: 60px;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

header h2 {
  color: white;
  margin-right: 40px;
}

header nav {
  margin-left: 40px;
  width: 30%;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

header nav a {
  list-style: none;
  text-decoration: none;
  color: white;
}

/*****content*****/

.content {
  height: 100vh;
  background: linear-gradient(rgba(0, 0, 0, 0.8), rgba(0, 0, 0, 0.8)), url("ima
}

.content h1 {
  color: white;
  text-align: center;
  padding-top: 250px;
  margin-bottom: 90px;
}

```

```
    letter-spacing: 1.4rem;
    font-size: 6rem;
    font-family: 'cormorant';
    font-weight: 100;
}

.content h2 {
    color: white;
    text-align: center;
    letter-spacing: 2rem;
    font-size: 4rem;
    font-family: 'cormorant';
    margin-top: 100px;
    font-weight: 100;
}

.content hr {
    margin: 0 auto;
    width: 20%;
}
```

Cliquez sur le fichier *index.html* afin de pouvoir visualiser le résultat dans votre navigateur Web

Il n'est pas question de comprendre l'ensemble du code ci-dessus, mais vous pouvez apporter des modifications à ce code (HTML et/ou CSS) afin de visualiser les effets de ces modifications. Vous pouvez aussi carrément supprimer le fichier CSS et visualiser les effets de cette suppression.

activité 23.2

Élaborez une page HTML en utilisant les balises obligatoirement les balises suivantes :

- h1
- h2
- p
- img
- a
- form
- label
- input
- button

Il faudra styliser votre page avec du CSS



Ce qu'il faut savoir

- Le HTML (HyperText Markup Language) est un langage de balisage conçu pour représenter des pages web.
- Dans une page HTML on trouve uniquement des balises ouvrantes (exemple `<p>`) et fermantes (exemple `</p>`) ou des balises à la fois ouvrantes et fermantes (exemple ``)
- Les navigateurs web (Firefox, Chrome...) sont des logiciels capables d'interpréter du HTML et d'afficher la page web correspondante.
- Chaque balise a une signification qu'il faut bien respecter (on parle de la sémantique des balises).
- Le HTML n'a pas été conçu pour gérer la mise en page. Le HTML s'occupe uniquement du contenu et de la sémantique, pour tout ce qui concerne la mise en page et l'aspect "décoratif" (on parle du "style" de la page), on utilisera un autre langage, le CSS (Cascading Style Sheets).
- il est possible d'ajouter des attributs dans les balises

Ce qu'il faut savoir faire

- Vous devez savoir utiliser un certain nombre de balises : la balise "paragraphe" `<p>` , les balises titre `<h1>` , `<h2>` ,..., la balise lien `<a>` , la balise image `` , la balise formulaire `<form>` , la balise `<input>` , la balise `<button>` et les balises génériques `<div>` et ``
- Vous devez à partir d'un code HTML et d'un code CSS (associé au code HTML) être capable de "deviner" l'aspect d'une page web (couleur d'un texte, police de caractère,...)